IN THE CLAIMS

Please amend the claims as follows:

Claim 1 (Currently Amended): A compiler system for generating object code from an input source program, comprising:

a character string interpreter configured to divide instructions coded within an input source program into tokens;

an intrinsics function information database into which a definition of an intrinsics function and an instruction attribute information characterizing an instruction coded in intrinsics function are stored as intrinsics function information;

a ~~first~~ syntax analyzer configured to analyze syntax of said tokens, ~~and~~ to judge as to whether or not a definition of an intrinsics function ~~and its kind of operand~~ is included in a combination of said tokens[[;]], ~~a second syntax analyzer configured~~ to find a reserved pre-processing instruction in the combination of said token and, if found, to add said instruction attribute information of said intrinsics function described in said pre-processing instruction to the definition of the intrinsics function in said database; and

a code generator which develops configured to develop an instruction that calls an intrinsics function within said source program by referring to said intrinsics function information, and to convert said developed source program either to machine language or to an intermediate code.

Claim 2 (Currently Amended): The compiler system according to claim 1, wherein said ~~first~~ syntax analyzer distinguishes a prescribed identifier that indicates an intrinsics function from among a function declaration part of said source program to judge as to whether or not said intrinsics function definition and said instruction attribute information is defined.

Claim 3 (Previously Presented): The compiler system according to claim 1, wherein said intrinsics function definition includes a dummy argument type and identification name.

Claim 4 (Currently Amended): A compiler system for generating object code from an input source code program, comprising:

a character string interpreter configured to divide instructions coded within an input source program into tokens;

an intrinsics function information database into which a definition of an intrinsics function and an instruction attribute information characterizing an instruction coded in intrinsics function are stored as intrinsics function information;

a ~~first~~ syntax analyzer configured to analyze syntax of said tokens, ~~and~~ to judge as to whether or not a definition of an intrinsics function ~~and its kind of operand~~ is included in a combination of said tokens[[;]], ~~a second syntax analyzer configured~~ to find a reserved pre-processing instruction in the combination of said token and, if found, to add said instruction attribute information of said intrinsics function described in said pre-processing instruction to the definition of the intrinsics function in said database; and

a code generator configured to develop an instruction that calls an intrinsics function within said source program by referring to said intrinsics function information, and to convert said developed program either to machine language or to an intermediate code.

Claim 5 (Currently Amended): The compiler system according to claim 4, wherein said ~~first~~ syntax analyzer distinguishes a prescribed identifier indicating an intrinsics function from among a function declaration part coded within an external file, thereby making a judgment as to whether or not an intrinsics function definition and said instruction attribute information are defined.

Claim 6 (Previously Presented): The compiler system according to claim 4, wherein said intrinsics function definition includes a dummy argument type and identification name.

Claim 7 (Currently Amended): A compiler system for generating object code from an input source program, comprising:

a character string interpreter configured to divide instructions coded within an input source program into tokens;

an intrinsics function information database into which a definition of an intrinsics function and an instruction attribute information characterizing an instruction coded in intrinsics function are stored as intrinsics function information;

a ~~first~~ syntax analyzer configured to analyze syntax of said tokens, ~~and~~ to judge as to whether or not a definition of an intrinsics function ~~and its kind of operand~~ is included in a combination of said tokens[[;]], ~~a second syntax analyzer configured~~ to find a reserved pre-processing instruction in the combination of said token and, if found, to add said instruction attribute information of said intrinsics function described in said pre-processing instruction to the definition of the intrinsics function in said database; and

a code generator configured to develop an instruction that calls an intrinsics function within said source program by referring to said intrinsics function information, and to convert said expanded source program either to machine language or to intermediate code,

wherein said intrinsics function information includes a function declaration statement, to which is added a prescribed identifier indicating an intrinsics function, dummy argument information, and said instruction attribute information.

Claim 8 (Currently Amended): A method for compiling which generates object code from an input source program, comprising:

storing a definition of an intrinsics function into an intrinsics function information database;

storing instruction attribute information characterizing an instruction coded by an intrinsics function into said intrinsics function information database;

dividing instructions coded within an input source program into tokens;

analyzing syntax of said tokens, and judging as to whether or not a definition of an intrinsics function and its kind of operand is included in a combination of said tokens;

finding a reserved pre-processing instruction in the combination of said token and, if found, adding said instruction attribute information of said intrinsics function described in said pre-processing instruction to the definition of the intrinsics function in said database; and

developing an instruction that calls an intrinsics function within said source program by referring to said intrinsics function information database, and converting said developed source program either to machine language or to intermediate code.

Claim 9 (Currently Amended): A method for compiling, which generates object code from an input source program, comprising:

dividing instructions coded within an input source program into tokens;

analyzing syntax of said tokens, and judging as to whether or not a definition of an intrinsics function and its kind of operand is included in a combination of said tokens;

finding a reserved pre-processing instruction in the combination of said token and, if found, adding said instruction attribute information of said intrinsics function described in said pre-processing instruction to the definition of the intrinsics function;

accessing an intrinsics function information database, into which are stored a definition of an intrinsics function and instruction attribute information characterizing an instruction coded by said intrinsics function;

developing an instruction that calls an intrinsics function within said source program; and

converting said developed source program either to machine language or to intermediate code.

Claim 10 (Previously Presented): The method for compiling according to claim 8, wherein said analyzing syntax of said tokens is performed by distinguishing a prescribed identifier indicating an intrinsics function from among a function declaration part of said source program, thereby making a judgment as to whether or not said intrinsics function definition and said instruction attribute information are defined.

Claim 11 (Previously Presented): The method for compiling according to claim 9, wherein said analyzing syntax of said tokens is performed by distinguishing a prescribed identifier indicating an intrinsics function from among a function declaration part of said source program, thereby making a judgment as to whether or not said intrinsics function definition and said instruction attribute information are defined.

Claim 12 (Currently Amended): A computer-readable recording medium onto which is stored a program causing a computer to execute compiling processing that generates object code from an input source program, said program comprising:

processing for character string interpretation, so as to divide instructions coded within an input source program into tokens;

processing for storing a definition of an intrinsics function and an instruction attribute information characterizing an instruction coded in intrinsics function as intrinsics function information;

processing for analyzing syntax of said tokens, and judging as to whether or not a definition of an intrinsics function and its kind of operand is included in a combination of said tokens;

processing for finding a reserved pre-processing instruction in the combination of said token and, if found, adding said instruction attribute information of said intrinsics function described in said pre-processing instruction to the definition of the intrinsics function;

processing for developing an instruction that calls said intrinsics function within said source program by referring to said intrinsics function information; and

generating code that converts said expanded source program either to machine language or to intermediate code.

Claim 13 (Original): The recording medium according to claim 12, wherein said syntax analysis processing distinguishes a prescribed identifier indicating an intrinsics function from among a function declaration part of said source program, thereby making a judgment as to whether or not said intrinsics function definition and said instruction attribute information are defined.

Claim 14 (Currently Amended): A computer-readable recording medium onto which is stored a program causing a computer to execute compiling processing that generates object code from an input source program, the program comprising:

processing for character string interpretation, so as to divide instructions coded within an input source program into tokens;

processing for storing a definition of an intrinsics function and instruction attribute information characterizing an instruction coded by said intrinsics function as intrinsics function information;

processing for analyzing syntax of said tokens, and judging as to whether or not a definition of an intrinsics function and its kind of operand is included in a combination of said tokens;

processing for finding a reserved pre-processing instruction in the combination of said token and, if found, adding said instruction attribute information of said intrinsics function described in said pre-processing instruction to the definition of the intrinsics function;

processing for developing an instruction that calls an intrinsics function within said source program by referring to said intrinsics function information; and

generating code that converts said developed source program either to machine language or to intermediate code.

Claim 15 (Original): The recording medium according to claim 14, wherein said syntax analysis processing distinguishes a prescribed identifier indicating an intrinsics function from among a function declaration part of said source program, thereby making a judgment as to whether or not said intrinsics function definition and said instruction attribute information are defined.

Claim 16 (Currently Amended): A computer-readable recording medium onto which is stored a program causing a computer to execute compiling processing that generates object code from an input source program, said program comprising:

processing for character string interpretation, so as to divide instructions coded within an input program into tokens;

processing for storing a definition of an intrinsics function and instruction attribute information characterizing an instruction coded by said intrinsics function as intrinsics function information;

processing for analyzing syntax of said tokens, and judging as to whether or not a definition of an intrinsics function and its kind of operand is included in a combination of said tokens;

processing for finding a reserved pre-processing instruction in the combination of said token and, if found, adding said instruction attribute information of said intrinsics function described in said pre-processing instruction to the definition of the intrinsics function;

processing for accessing said intrinsics function information;

processing for developing an instruction that calls an intrinsics function within said source program by referring to said intrinsics function information; and

processing for generating code that converts the thus developed source program either to machine language or to intermediate code,

wherein said intrinsics function information comprises a function declaration statement to which is added a prescribed identifier indicating an intrinsics function, dummy argument information, and said instruction attribute information.

Claim 17 (Currently Amended): A program product for causing a computer to execute a program for generating object code from an input source program, said program comprising:

a process for character string interpretation so as to divide instructions coded within an input source program into tokens;

a process for analyzing syntax of said tokens, and judging as to whether or not a definition of an intrinsics function and its kind of operand is included in a combination of said tokens;

a process for finding a reserved pre-processing instruction in the combination of said token and, if found, adding instruction attribute information of said intrinsics function described in said pre-processing instruction to the definition of the intrinsics function;

a process for storing said intrinsics function definition and said instruction attribute information as intrinsics function information;

a process for developing an instruction that calls an intrinsics function within said source program by referring to said intrinsics function information; and

a process for generating code that converts said developed source program either to machine language or to intermediate code.

Claim 18 (Previously Presented): The program product according to claim 17, wherein said syntax analysis process distinguishes a prescribed identifier that indicates an intrinsics function from among a function declaration part of said source program, thereby making a judgment as to whether or not said intrinsics function definition and said instruction attribute information are defined.

Claim 19 (Currently Amended): A program product for causing a computer to execute a program for generating object code from an input source program, said program comprising:

a process for character string interpretation so as to divide instructions coded within an input source program into tokens;

a process for storing an intrinsics function definition and attribute information characterizing an instruction coded by said intrinsics function as intrinsics function information;

a process for analyzing syntax of said tokens, and judging as to whether or not a definition of an intrinsics function and its kind of operand is included in a combination of said tokens;

a process for finding a reserved pre-processing instruction in the combination of said token and, if found, adding said instruction attribute information of said intrinsics function described in said pre-processing instruction to the definition of the intrinsics function;

a process for accessing said intrinsics function information;

a process for developing an instruction that calls an intrinsics function within said source program by referring to said intrinsics function information; and

a process for generating code that converts said developed source program either to machine language or to intermediate code.

Claim 20 (Currently Amended): A program product for causing a computer to execute a program for generating object code from an input source program, said program comprising:

a process for character string interpretation so as to divide instructions coded within a source program into tokens;

a process for storing an intrinsics function definition and attribute information characterizing an instruction coded by said intrinsics function as intrinsics function information;

a process for analyzing syntax of said tokens, and judging as to whether or not a definition of an intrinsics function ~~and its kind of operand~~ is included in a combination of said tokens;

a process for finding a reserved pre-processing instruction in the combination of said token and, if found, adding said instruction attribute information of said intrinsics function described in said pre-processing instruction to the definition of the intrinsics function;

a process for accessing said intrinsics function information;

a process for developing an instruction that calls an intrinsics function within said source program by referring to said intrinsics function information; and a process for generating code that converts said developed source program either to machine language or to intermediate code,

wherein said intrinsics function information is made up of a function declaration statement to which is added a prescribed identifier indicating an intrinsics function, dummy argument information, and said instruction attribute information.